L2/02-283

Myanmar Script Canonical Ordering

Martin Hosken,
SIL Non-Roman Script Initiative (NRSI)

Abstract

This paper presents a possible canonical ordering for the Unicode representation of Myanmar script. It examines the ordering with reference to various processes including rendering, keying, sorting, clustering, transcription and transliteration. It also includes an executive summary in the form of a suggested text for inclusion in the Unicode Standard description for the Myanmar script.

Executive Summary

The following text is proposed for inclusion in the Unicode Standard as part of the description of the Myanmar script, replacing the corresponding paragraph which is currently there.

Signs After Consonants. Dependent vowels and other signs are encoded after the consonant to which they apply, apart from kinzi which occurs before. Characters occur in the following relative order:

Name	Specification	Example
kinzi	U+1004 U+1039	◌
Consonant	[U+1000 .. U+1021]	◌
Stacked	U+1039 [U+1000 .. U+1019, U+101C, U+101E, U+1020, U+1021]	◌
Medial H	U+1039 U+101F	◌
Medial Y	U+1039 U+101A	◌
Medial R	U+1039 U+101B	◌
Medial W	U+1039 U+101D	◌
E vowel	U+1031	◌
Lower Vowel	[U+102F, U+1030]	◌
Upper Vowel	[U+102D, U+102E, U+1032]	◌
A Vowel	U+102C	◌
Anusvara	U+1036	◌
Visible killer	U+1039 U+200C	◌
Lower Dot	U+1037	◌
Visarga	U+1038	◌

Introduction

In addition to providing a set of properties for each character code, it is necessary to specify in which order non-spacing characters should occur relative to their base character. This is called the canonical ordering for the script. For many scripts, this specification is relatively straightforward. For the Myanmar script, this specification is more complex. There are various processes that a text may be

put through including: clustering, rendering, keying, sorting, transcription and transliteration. Whereas, for most scripts, the most appropriate canonical ordering of the text is similar for most of the processes, for the Myanmar script, each of the processes implies a different canonical ordering. This, combined with the number of characters involved leads to great care needing to be taken when establishing an underlying canonical ordering for the Myanmar script.

The need for such an ordering is essential for implementors. Without an agreed convention on the canonical ordering of data, each implementation will need to be able to work with any random ordering which may present itself. Strings which look the same will not match as being the same and implementations will be hard to come by. Establishing this order now will free the industry to work on implementations and spend its time producing solutions that users want without having to cover the same ground every time.

Compound Clusters

In describing the canonical ordering for the Myanmar script, we need to consider the order in terms of what constitutes a compound cluster of characters. This compound cluster is akin to a grapheme cluster[1], but as we shall see, can itself be broken down into grapheme clusters. The compound cluster is also akin to a syllable, but since the sequence ၍ constitutes a compound cluster, it is clear that it may take more than one compound cluster to describe a syllable.

Regular Expressions

As an aid to describing the various orderings, we introduce a list of character classes. Each class specifies a set of characters or short character sequences. A compound cluster is described using a simple regular expression language to describe all the possible character sequences which conform to the specified order. Notice that it is not the purpose of such descriptions to stipulate what characters may or may not occur with each other, but to specify their relative order.

Identifier	Name	Specification	Examples
G	Ng prefix	U+1004 U+1039	◌
C	Consonant	[U+1000 .. U+1021]	◌
M	Stacked	U+1039 [U+1000 .. U+1019, U+101C, U+101E, U+1020, U+1021]	◌
H	Medial H	U+1039 U+101F	◌
R	Medial R	U+1039 U+101B	◌
Y	Medial Y	U+1039 U+101A	◌
W	Medial W	U+1039 U+101D	◌
E	E vowel	U+1031	◌
L	Lower Vowel	[U+102F, U+1030]	◌
U	Upper Vowel	[U+102D, U+102E, U+1032]	◌
A	A Vowel	U+102C	◌
S	Anusvara	U+1036	◌
K	Visible killer	U+1039 U+200C	◌
D	Lower Dot	U+1037	◌
V	Visarga	U+1038	◌

[1] A grapheme cluster, or simply cluster, is a visually irreducible character sequence. In an editor, for example, it would not be possible to insert a cursor between characters within a grapheme cluster.

In the class specifications, [ ] indicates a set of characters, with . . indicating a range of possible values. Thus the M class can be read as being a virama (U+1039) followed by any character in the range U+1000 to U+1019 or U+101C or U+101E or U+1020.

In terms of a regular expression, we might describe the canonical ordering as currently specified in the Maynmar script rubric in the Unicode Standard v3.0 as being covered by:

```
G? C Y? W? E?
```

That is that there is an optional Ng prefix followed by a consonant followed by an optional medial Y and then an optional medial W and then an optional E vowel. Notice that while each of the elements are optional, if any of them are present, the expression specifies what their relative order must be. Thus the E vowel may not occur before the Consonant. Likewise the medial W may not occur before the medial Y and the Ng prefix must occur before the Consonant and not after it. Notice that the main consonant (C) is not optional. Every compound cluster requires a main consonant.

Basic Principles

The primary language of interest when considering the Myanmar script, is Burmese. The Burmese syllable is structured as an onset (consonant) followed by a rhyme. The rhyme contains the vowel, final consonant and tone. This provides an overall canonical order for Myanmar script. It may be described using one or more compound clusters with the first consisting of the onset and vowel and even the final. Another syllable may use two compound clusters, the first containing the onset and vowel and the second the final and tone.

The rest of the discussion is concerned with the details within that overall ordering. For example, when two vowel characters occur, should the upper vowel precede the lower vowel, or vice versa? At this level, the decisions are somewhat arbitrary, since the user should never need to know what the relative order is. Having said this, the choice is of technical interest for implementors since the relative order can aid or hinder their development work. In addition, a poor implementation may result in the user having to be aware of the underlying canonical ordering.

Processes

In this section we examine each of the processes that a text in the Myanmar script may need to go through. The processes are examined in an order suited to building up a description of the canonical order.

Transcription

Transcription is the process of representing the sound of a text in a language represented in one script, in another script. Thus the purpose is not necessarily to provide a round-trip equivalence. The result is that the transcription process is somewhat akin to the reading process and from it we can extract a reading based canonical order. In the case of Burmese, transcription is a common process, but is particularly problematic due to various irregular spellings. The approach taken here is script based and cross-linguistic. Any real transcription process for the Burmese language would have to deal with all the irregularities.

By examining the transcription process we can arrive at a relative order for the medial consonants. For the most part only one medial is ever used, but there are occasions where 2 occur together. While not every combination of 2 medials occurs, there are enough combinations that a relative order can be arrived at. This is:

```
G? C M? H? Y? R? W?
```

Thus a consonant may be followed by a syllable chained letter (i.e. the first consonant is really part of the previous syllable and the M consonant is the start of this syllable) and then by any number of medial letters in the given order. Notice that there are no occasions where all five characters are used together. The purpose here is not to specify a restriction, but to specify a relative ordering.

There are various words which show that the above order is the appropriate one: ရွှေ /shwe/[2] 'gold' (H before W), မျိုစ် /hmyiq/ 'bamboo shoot' (H before Y), ကျွန်း /cun/ 'island' (Y before W), ငြွေး /cwe/ 'debt' (R before W), ညှိုင်း /hnyein/ 'to extinguish' (H before R), မြှော /hmywa/ 'to segmentalize' (H before R before W).

During transcription, vowel sequences are resolved into their combined sound, which does not provide for a breakdown in the vowel components.

Transliteration

Transliteration is the process of spelling the letters in one script in another script. Thus the purpose here is that orthographic differences be reflected in the target script, rather than primarily being concerned with representing the original sounds. Transliteration schemes, are by their nature, artificial. But, since they aim to communicate, they exist as standards within their user communities. Can they help in resolving some of the ordering ambiguities?

The primary remaining concern is regarding the /o/ vowel: ◌ိ◌ whether it should be stored: ◌◌ိ ◌ိ◌ or ◌ိ◌◌. There seems to be universal agreement that this vowel is transliterated /ui/. This would imply that the second ordering is the correct one. Thus we arrive at the following order for consonant clusters and vowels:

```
G? C M? H? Y? R? W? E? L? U?
```

Continuing with the rest of the compound cluster we get:

```
G? C M? H? Y? R? W? E? L? U? A? S? K? D? V?
```

With D and V being tone marks and S and K as finals, we have something which conforms to the general structure of consonant, medials, vowels, final, tone.

Clustering

There are a number of places where it might be preferable to allow a cursor to be placed within a compound cluster. In the list of classes, C, E, A and V are all spacing. All other classes are non-spacing. The compound cluster may constitute up to 3 grapheme clusters: a main cluster including C and E and all non-spacing characters up to A, a cluster starting with A and its non-spacing diacritics and finally with a cluster consisting of V. Notice that a syllable may well consist of more than one compound cluster, since any killed consonant, and its corresponding tone mark, is part of the preceding syllable. This can result in a visual chaining of syllables, where it is not possible to graphically break the syllables apart for line breaking purposes, for example.

Examining the order, we notice that the E is positioned right in the middle of the main cluster. There is nothing that can be done about this, and since E needs special treatment for rendering purposes anyway, that same special treatment can be used for cursor location as well.

Rendering

The most obvious processing need for a text is that of rendering. Since the Myanmar script works to a virama model, it is necessary to use smart font technology to re-organise the glyphs appropriately for

[2] Transcription according to Okell 1994, Appendix 2.

rendering. Once that requirement is made, the level of sophistication of modern smart font rendering technologies is high and amply sufficient to meet the needs of re-ordering elements to render a string with diacritics occurring in the above order. The re-ordering of a smart font might result in the following order. Notice that this order is only used internally to the rendering process and would not result in data being stored in this order.

```
E? R? C G? M? H? W? Y? L? U? S? A? K? D? V?
```

This involves four slots moving. Even if the basic order were adjusted to still keep the same basic order while optimising the sequence for rendering, three slots would still need to move. Therefore, it is not recommended that the canonical order be optimised for rendering.

Keying

Probably the most complex area for an implementation will be keying. There are so many different orders that users will want to be able to type. Users often type the /o/ vowel (◌ုိ) in either of the two orders; even to the extent of changing the typing order within the same document. Therefore, again, it is not wise to base the canonical ordering on one particular user's desires regarding keying order.

Even if an implementation demands that a user key data in canonical order only, this should not result in too obscure a notion of keying, even if this will frustrate users initially. For example, many users will be used to typing an E vowel (◌ေ) before the consonant it sounds after. But the underlying order requires that the E vowel must follow the consonant. Solutions geared towards what users are used to now, will require re-ordering of data from keying order to canonical order. This could well be the difference between a poor and excellent script implementation.

Sorting

As it stands the canonical order listed here can result in a Pali based sort without need for re-ordering. Modern spelling book ordering is based on the final consonant having higher priority than the vowel. To support the spelling book order directly, it would be necessary to re-organise the canonical ordering to reverse the rhyme and have the final consonant precede the vowel. This would cause sufficient confusion in all other areas as to make it not worthwhile. Instead, spelling book order may either be achieved via a pre-processing pass to get the characters into a suitable order or using a sorting process in which all the rhymes are listed as their own collation elements.

Bibliography

Okell, John *Burmese: An Introduction to the Script* (Northern Illinois University: 1994)

Roop, H. D. *An Introduction to the Burmese Writing System* (New Haven, Yal University Press: 1972)